4bears Technologies

Dino Explorer

MVS Data Collector for Mainframe Assessment

Index of contents

1 - Introduction 3
2 - The Installation Guide 4
2.1 – Step 1 – Copy the dxplload.XMIT to MVS 4
2.2 – Step 2 – Receive XMIT file 6
3 - Running the Data Collectors.....8
3.1 – DXPLSMF - SMF Extractor 8
3.2 – DXCOLET – DCOLLECT Extractor 11
3.3 - Downloading the CSV files.....12

1 - Introduction

This document summarize the procedures for the installation and the execution of the data collectors utilities to gather the required data from MVS systems to be used on the Mainframe Assessment.

The data collector utilities are part of the Dino Explorer Suite, and detailed information can be found on the Dino Explorer MVS Installation Guide manual.

Currently, we use the following data:

Data	Source	Collector utility
SMF 14/15 (DS)	SMF dump	DXPLSMF
SMF 30 (CPU)	SMF dump	DXPLSMF
SMF 42.6 (SMF)	SMF dump	DXPLSMF
SMF 64 (VSAM)	SMF dump	DXPLSMF
RMF 70.1 (CEC)	SMF dump	DXPLSMF
RMF 73 (CHPID)	SMF dump	DXPLSMF
RMF 74.1 (DEVICE)	SMF dump	DXPLSMF
RMF 74.5 (CACHE)	SMF dump	DXPLSMF
RMF 78.3 (LCU)	SMF dump	DXPLSMF
DCOLLECT Volume (V) records	IDCAMS	DXCOLET

The installation process is very simple, consisting of the upload of a load-library in XMIT format (DXPL.V400.DXPLLOAD.XMIT). This process is described on the Chapter 2.

The collector utilities generate a single output file in text format (CSV), that should be downloaded from the mainframe in text mode (ASCII) or if you prefer, you can compact the file with TRSMAIN (terse) and transfer the tersed file in binary mode (BIN). Please refer to Chapter 3 for instructions.

The DXPLSMF can read your SMF tapes directly and you can generate as many output CSV files as is convenient to you.

2 - The Installation Guide

To install the DXPLLOAD library on the mainframe you need just execute these two steps:

Step	Description of SMF Collector installation
1	Copy the dxplload in XMIT format to MVS
2	Receive the dxplload to restore to library format (PDSE)

The only dataset the product requires is the load library that we refer as:

DXPL.V400.DXPLLOAD

2.1 – Step 1 – Copy the dxplload.XMIT to MVS

You receive the dxplload in a XMIT format that is a special format used by IBM to transfer files between systems. The file is DXPL.V400.DXPLLOAD.XMIT

Pre-allocate the XMIT file on MVS

The easiest way to deal with XMIT files is pre-allocating the file in the MVS before transferring the XMIT file because you do not have to enter any DCB information on the FTP command. So, we pre-allocate a XMIT file on the mainframe:

```
                          Data Set Information

Data Set Name  . . . : DXPL.V400.DXPLLOAD.XMIT

General Data                          Current Allocation
 Volume serial . . . : D4BEAR          Allocated cylinders : 1
 Device type . . . . : 3390            Allocated extents . : 1
 Organization  . . . : PS
 Record format . . . : FB
 Record length . . . : 80
 Block size  . . . . : 3120           Current Utilization
 1st extent cylinders: 5               Used cylinders  . . : 0
 Secondary cylinders : 1               Used extents  . . . : 0
```

Transfer the XMIT to MVS

You can use FTP to copy the dxplload.xmit to the dataset you have just allocated. Do not forget to set binary mode (bin).

Open a command prompt on the windows (cmd.exe) and type the following

1 Go to the directory where the xmit file are:

```
C:\>cd \temp
C:\temp>
```

2 Connect to the mainframe FTP and supply an user ID and password as requested:

```
C:\temp>ftp 10.5.0.2
Connected to 10.5.0.2.
220-FTPD1 IBM FTP CS V2R10 at S390.dino.br, 11:43:19 on 2008-02-21.
220 Connection will close if idle for more than 5 minutes.
User (10.5.0.2:(none)):
```

3 Change to root (cd ..) and set binary mode (bin) entering the commands:

```
ftp> cd ..
250 "" is the working directory name prefix.
ftp> bin
200 Representation type is Image
ftp>
```

4 Transmit the xmit file:

```
ftp> put DXPL.V400.DXPLLOAD.XMIT
200 Port request OK.
125 Storing data set DXPL.V400.DXPLLOAD.XMIT
250 Transfer completed successfully.
ftp: 1988000 bytes sent in 9,40Seconds 211,51Kbytes/sec.
ftp>
```

2.2 – Step 2 – Receive XMIT file

Once you have transferred the dxplload in XMIT format to the MVS, we need to convert it to the library format using the TSO RECEIVE command.

It is advisable to pre-allocate the dxplload library in a permanent disk (not work) before issuing the RECEIVE command.

Pre-allocate the dxplload library

You can pre-allocate the dxplload using the following parameters:

```
                          Allocate New Data Set
                                                                More:
Data Set Name  . . . : DXPL.V400.DXPLLOAD

Management class . . .                 (Blank for default management class)
Storage class  . . . .                 (Blank for default storage class)
 Volume serial . . . . D4BEAR          (Blank for system default volume) **
 Device type . . . . .                 (Generic unit or device address) **
Data class . . . . . .                 (Blank for default data class)
 Space units . . . . . CYLINDER        (BLKS, TRKS, CYLS, KB, MB, BYTES
                                        or RECORDS)
 Average record unit                   (M, K, or U)
 Primary quantity  . . 5               (In above units)
 Secondary quantity    5               (In above units)
 Directory blocks  . . 0               (Zero for sequential data set) *
 Record format . . . . U
 Record length . . . . 0
 Block size  . . . . . 27000
 Data set name type  : LIBRARY         (LIBRARY, HFS, PDS, or blank)  *
                                       (YY/MM/DD, YYYY/MM/DD
 Expiration date . . .                  YY.DDD, YYYY.DDD in Julian form
```

Receive the dxplload

Now, we can receive the dxplload, i.e. to convert from XMIT format to the load library format on the pre-allocated library:

On the TSO command on the ISPF (6 Command):

```
RECEIVE INDA('DXPL.V400.DXPLLOAD.XMIT')
```

Menu List Mode Functions Utilities Help
ISPF Command Shell
Enter TSO or Workstation commands below:
===> RECEIVE INDA('DXPL.V400.DXPLLOAD.XMIT')

```
INMR901I Dataset DXPL.V400.DXPLLOAD from GAZOLA on NODENAME
INMR154I The incoming data set is a 'PROGRAM LIBRARY'.
INMR906A Enter restore parameters or 'DELETE' or 'END' +
```

If you leave the dsname of the dxplload as the original: DXPL.V400.DXPLLOAD, you just need press <enter> to confirm the command. Otherwise you enter the new name as bellow:

```
DA('NEWNAME.DXPLLOAD')
```

```
INMR045I Dataset 'DXPL.V400.DXPLLOAD' already exists.  Reply 'R' to replace
identically named members. +
***
```

Enter 'R' and <enter> to confirm the replace because the dxplload was already pre-allocated:

```
 IGW01551I MEMBER DXPLCCMD HAS BEEN LOADED
 IGW01551I MEMBER DXPLDASD HAS BEEN LOADED
 IGW01551I MEMBER DXPLDINO HAS BEEN LOADED
 IGW01551I MEMBER DXPLSMF  HAS BEEN LOADED
 IGW01551I MEMBER DXPLSSI  HAS BEEN LOADED
 IGW01551I MEMBER DXPLTCP  HAS BEEN LOADED
 IGW01551I MEMBER PARMSMF  HAS BEEN LOADED
 IGW01550I 7 OF 7  MEMBERS WERE LOADED
 IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE
 INMR001I Restore successful to dataset 'DXPL.V400.DXPLLOAD'
 ***
```

3 - Running the Data Collectors

This part explains how to run the data collector utilities.

3.1 – DXPLSMF - SMF Extractor

The DXPLSMF utility reads a SMF dump file and extracts the relevant information from the SMF records and written the output on a text file in CSV format, the CSVOUT ddname.

SMF records

The following table describe the records required by each product:

Option	Product	SMF records
CACHE	DASD Explorer	Type 74 subtype 5 - CACHE Activity
CAT	Dataset Explorer	Type 61 - ICF Define Activity Type 65 - ICF Delete Activity Type 66 - ICF Alter Activity
CEC	CPU Explorer	Type 70 subtype 1 - RMF Processor Activity
CHPID	DASD Explorer	Type 73 - RMF CHANNEL PATH Activity
CPU	CPU Explorer	Type 30 subtypes 1 to 6
DEVICE	DASD Explorer	Type 74 subtype 1 - RMF DEVICE Activity
DS	Dataset Explorer	Type 14 – Non-VSAM input dataset Type 15 – Non-VSAM output dataset
IO	IO Explorer	Type 30 subtype 4 – Step termination
LCU	DASD Explorer	Type 78 subtype 3 - RMF I/O QUEUEING Activity
SMS	Dataset Explorer	Type 42 subtype 6 - SMS I/O statistics
VS	Dataset Explorer	Type 64 – VSAM close

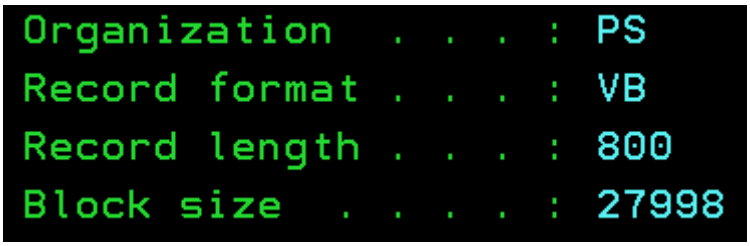
JCL

The following JCL can be used to collect the SMF data:

```
//DXPLSMF  JOB MSGLEVEL=(1,1),NOTIFY=&SYSUID
//DXPLSMF  EXEC PGM=DXPLSMF
//STEPLIB  DD DISP=SHR,DSN=DXPL.V400.DXPLLOAD
//SMFIN    DD DISP=SHR,DSN=smf.dump.file
//CSVOUT   DD DSN=dxpl.csvout,UNIT=SYSALLDA,
//         DISP=(NEW,CATLG),VOL=SER=volser,
//         SPACE=(CYL,(100,50),RLSE)
//SYSPRINT DD SYSOUT=*
//DXPLIN   DD *
PROD CACHE
PROD CEC
PROD CPU
PROD CHPID
PROD DEVICE
PROD DS
PROD LCU
PROD VSAM
```

The DXPLSMF read the dump SMF file informed on the SMFIN DD and write the selected records on the CSVOUT dataset.

The CSV output files have the following format:



DXPLIN

You can specify some commands on the DXPLIN DD sysin accordingly to the following commands:

Command	Description
PROD ALL	Enable all product options.
PROD CACHE	Collects CACHE activity records.
PROD CAT	Collects ICF catalogs activity records.
PROD CEC	Collects RMF processor performance records.
PROD CHPID	Collects RMF Channel activity information.
PROD CPU	Collects data for the CPU Explorer.
PROD DEV	Collects RMF device activity records.
PROD DS	Collects non-VSAM data for the Dataset Explorer
PROD IO	Collects data for the IO Explorer.
PROD LCU	Collects LCU and Hyper-PAV information.
PROD SMS	Collects SMS Dataset I/O activity records.
PROD VS	Collects VSAM data for the Dataset Explorer
START dd/mm/yyyy hh:mm:ss	Collect only data with SMF recording time after the informed date and time.
END dd/mm/yyyy hh:mm:ss	Collect only data with SMF recording date and time before this date and time.

3.2 – DXCOLET – DCOLLECT Extractor

The DXCOLET utility is a batch program to extract information provided by the DFSMS Data Collection Facility (DCOLLECT). The DCOLLECT functions are obtained through the execution of the IDCAMS utility which are described on the DFSMS Access Method Services for Catalogs manual.

```
//DXCOLET  JOB  MSGLEVEL=(1,1),NOTIFY=&SYSUID
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=A
//OUTDS    DD   DSN=DXPL dcollect.output.file,
//         DSORG=PS,
//         DCB=(RECFM=VB,LRECL=644,BLKSIZE=0),
//         SPACE=(50,(100,100)),AVGREC=K,
//         DISP=(NEW,CATLG,KEEP)
//SYSIN    DD   *
           DCOLLECT  OFILE(OUTDS) VOLUME(**) NODATAINFO
//DXCOLET  EXEC PGM=DXCOLET,PARM='VOL,**'
//STEPLIB  DD   DISP=SHR,DSN=DXPL.V400.DXPLLOAD
//DCOLIN   DD   DISP=SHR,DSN=dcollect.output.file
//CSVFILE  DD    DSN=dxpl.csvfile,UNIT=SYSALLDA,
//         DISP=(NEW,CATLG),VOL=SER=volser,
//         SPACE=(CYL,(10,50),RLSE)
//SYSPRINT DD SYSOUT=*
```

The first step runs the IDCAMS utility to generate the OUTDS dataset with the DCOLLECT volume records that will be read by the second step (DXCOLET) which in turn will write the records on a CSV format on the CSVFILE ddname.

The CSVFILE should be transferred in text mode to the open platform.

3.3 - Downloading the CSV files

The output of the extractors are text files in CSV format (comma separeted values) and they are processed outside the mainframe on a system running Microsoft Windows. Text files in mainframes are EBCDIC code while on the open platform are ASCII code.

Depending on the site, the size of these files may be very large, so it is important to pay attention to the file transfer mode used to avoid retransmissions. We currently support the following transfer mode:

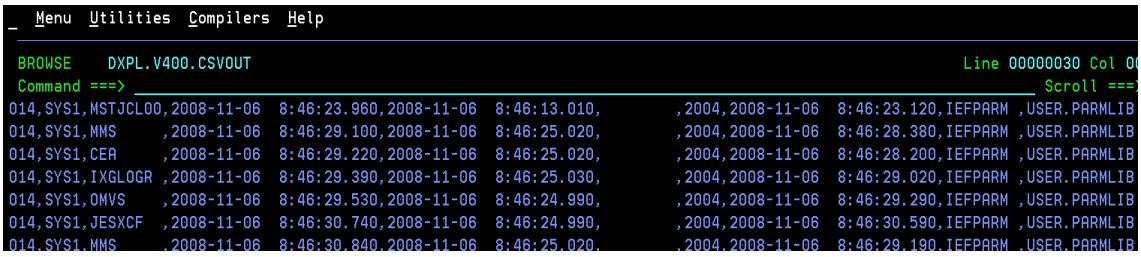
- Download the CSV files in TEXT mode
- Terse the CSV files on the mainframe and download them in BINARY mode

If you have good file transfer facility, it is better download the files in TEXT mode. You save mainframe CPU cycles. On the other hand, you can compress the file on the mainframe (TERSE), and download a file much smaller.

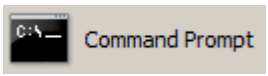
Method 1 - Downloading CSV files

The CSV files are transfered to the open platform in TEXT mode, i.e. the CSV file will be converted from mainframe EBCDIC code to ASCII code.

This is our prefered method because you can check the file contents any time. On the mainframe it looks like the picture bellow:



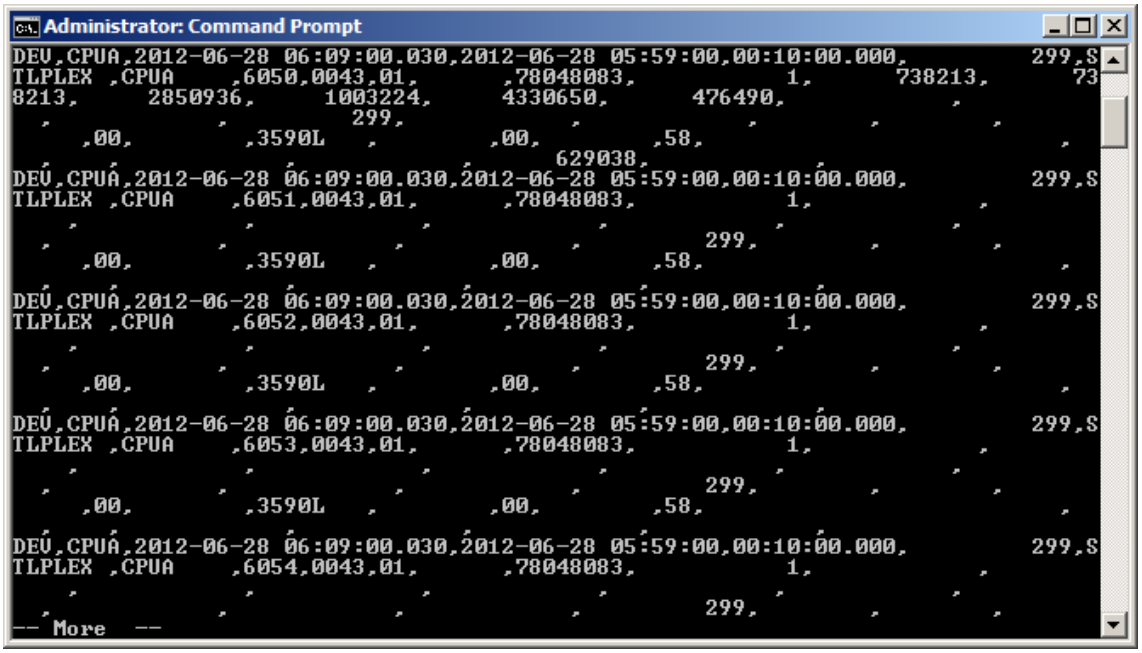
Note: Do not try to open the CSV file the Windows platform because it may take a long time to open the file. Intead, use the **type** with the **more** commands on a CMD command line:




```
C:\>cd \temp

C:\temp>type csvfile.csv | more
```

and you get an output such as:



After you transfer the file to the open platform, you can zip the files.

Method 2 - TERSE the CSV on the mainframe

You can compress the CSV files on the mainframe before transfering it to the open platform using the TRSMAIN utility.

The following example shows how to TERSE a file on the z/OS:

```
//DXTERSE   JOB MSGLEVEL=(1,1),NOTIFY=&SYSUID
//STEPNAME  EXEC PGM=TRSMAIN,PARM='PACK'
//SYSPRINT  DD SYSOUT=*
//INFILE    DD DISP=SHR,DSN= dxpl.csvfile
//OUTFILE   DD DISP=(,CATLG,DELETE),SPACE=(CYL,(100,500),RLSE),
//           DSN= dxpl.csvfile.trs,VOL=SER=volser,UNIT=3390
```

Note: You have to transfer the tersed file to the open platform in BINARY mode to preserve its format.